BSE658_Assign_

2024-10-29

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

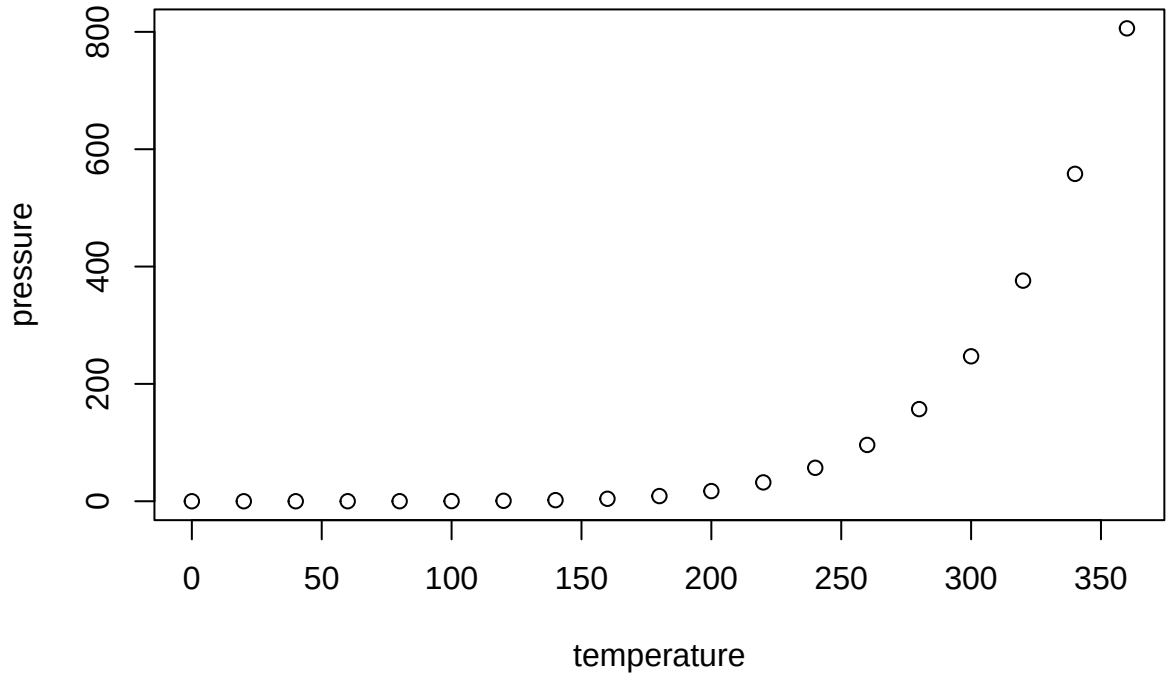
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

```r
# Load necessary libraries
library(lsr)
library(chisq.posthoc.test)
library(MKinfer)
library(psych)
```

```r
data("HairEyeColor")  # Load HairEyeColor dataset
str(HairEyeColor)
```

```
##  'table' num [1:4, 1:4, 1:2] 32 53 10 3 11 50 10 30 10 25 ...
##  - attr(*, "dimnames")=List of 3
##   ..$ Hair: chr [1:4] "Black" "Brown" "Red" "Blond"
##   ..$ Eye : chr [1:4] "Brown" "Blue" "Hazel" "Green"
##   ..$ Sex : chr [1:2] "Male" "Female"
```

```r
hair_eye_df <- as.data.frame(HairEyeColor)

# Observed frequencies for the 'Hair' variable among females
observed <- table(hair_eye_df$Hair[hair_eye_df$Sex == "Female"])
print(observed)
```

```
##
```

```
## Black Brown   Red Blond
##     4     4     4     4
```

```r
# Set expected probabilities (assuming equal distribution)
probabilities <- rep(1/length(observed), length(observed))
N <- sum(observed)
expected <- N * probabilities
print(expected)
```

```
## [1] 4 4 4 4
```

```r
# Chi-Square Goodness of Fit Test (manual calculation)
chi_square_calc <- sum((observed - expected)^2 / expected)
print(chi_square_calc)
```

```
## [1] 0
```

```r
# Goodness of Fit Test using goodnessOfFitTest function
goodnessOfFitTest(hair_eye_df$Hair[hair_eye_df$Sex == "Female"])
```

```
## Warning in goodnessOfFitTest(hair_eye_df$Hair[hair_eye_df$Sex == "Female"]):
## Expected frequencies too small: chi-squared approximation may be incorrect

##
##      Chi-square test against specified probabilities
##
## Data variable:   hair_eye_df$Hair[hair_eye_df$Sex == "Female"]
##
## Hypotheses:
##    null:        true probabilities are as specified
##    alternative: true probabilities differ from those specified
##
## Descriptives:
##       observed freq. expected freq. specified prob.
## Black             4              4            0.25
## Brown             4              4            0.25
## Red               4              4            0.25
## Blond             4              4            0.25
##
## Test results:
##    X-squared statistic:  0
##    degrees of freedom:  3
##    p-value:  1
##    warning: expected frequencies too small, results may be inaccurate
```

```r
# Chi-Square Goodness of Fit Test using chisq.test function
chisq.test(x = observed)
```

```
## Warning in chisq.test(x = observed): Chi-squared approximation may be incorrect
```

```
##
##  Chi-squared test for given probabilities
##
## data:  observed
## X-squared = 0, df = 3, p-value = 1
```

```
data("Titanic")  # Load Titanic dataset
str(Titanic)
```

```
##  'table' num [1:4, 1:2, 1:2, 1:2] 0 0 35 0 0 0 17 0 118 154 ...
##  - attr(*, "dimnames")=List of 4
##   ..$ Class   : chr [1:4] "1st" "2nd" "3rd" "Crew"
##   ..$ Sex     : chr [1:2] "Male" "Female"
##   ..$ Age     : chr [1:2] "Child" "Adult"
##   ..$ Survived: chr [1:2] "No" "Yes"
```

```
# Create a contingency table for class and survival
titanicFrequencies <- xtabs(Freq ~ Class + Survived, data = Titanic)
print(titanicFrequencies)
```

```
##       Survived
## Class   No Yes
##   1st  122 203
##   2nd  167 118
##   3rd  528 178
##   Crew 673 212
```

```
# Association Test for Class and Survival
associationTest(~ Class + Survived, data = as.data.frame(Titanic))
```

```
## Warning in associationTest(~Class + Survived, data = as.data.frame(Titanic)):
## Expected frequencies too small: chi-squared approximation may be incorrect
```

```
##
##      Chi-square test of categorical association
##
## Variables:   Class, Survived
##
## Hypotheses:
##    null:        variables are independent of one another
##    alternative: some contingency exists between variables
##
## Observed contingency table:
##       Survived
## Class  No Yes
##   1st   4   4
##   2nd   4   4
##   3rd   4   4
##   Crew  4   4
##
## Expected contingency table under the null hypothesis:
##       Survived
```

```
## Class  No Yes
##   1st   4   4
##   2nd   4   4
##   3rd   4   4
##   Crew  4   4
##
## Test results:
##    X-squared statistic:  0
##    degrees of freedom:  3
##    p-value:  1
##
## Other information:
##    estimated effect size (Cramer's v):  0
##    warning: expected frequencies too small, results may be inaccurate
```

```
chisq.test(titanicFrequencies)
```

```
##
##  Pearson's Chi-squared test
##
## data:  titanicFrequencies
## X-squared = 190.4, df = 3, p-value < 2.2e-16
```

```
# Post-hoc test
chisq.posthoc.test(titanicFrequencies)
```

```
##   Dimension     Value         No       Yes
## 1       1st Residuals -12.593038 12.593038
## 2       1st  p values   0.000000  0.000000
## 3       2nd Residuals  -3.521022  3.521022
## 4       2nd  p values   0.003439  0.003439
## 5       3rd Residuals   4.888701 -4.888701
## 6       3rd  p values   0.000008  0.000008
## 7      Crew Residuals   6.868541 -6.868541
## 8      Crew  p values   0.000000  0.000000
```

```
# Cramér's V for effect size
cramersV(titanicFrequencies)
```

```
## [1] 0.2941201
```

```
data("UCBAdmissions")  # Load UCBAdmissions dataset
str(UCBAdmissions)
```

```
##  'table' num [1:2, 1:2, 1:6] 512 313 89 19 353 207 17 8 120 205 ...
##  - attr(*, "dimnames")=List of 3
##   ..$ Admit : chr [1:2] "Admitted" "Rejected"
##   ..$ Gender: chr [1:2] "Male" "Female"
##   ..$ Dept  : chr [1:6] "A" "B" "C" "D" ...
```

```r
# Observed frequencies for 'Gender' and 'Admit'
admit_table <- margin.table(UCBAdmissions, c("Gender", "Admit"))
print(admit_table)
```

```
##         Admit
## Gender   Admitted Rejected
##   Male       1198     1493
##   Female      557     1278
```

```r
# Chi-Square and Fisher's Exact Test for Gender and Admission
chisq.test(admit_table)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  admit_table
## X-squared = 91.61, df = 1, p-value < 2.2e-16
```

```r
fisher.test(admit_table)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  admit_table
## p-value < 2.2e-16
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  1.621356 2.091246
## sample estimates:
## odds ratio
##   1.840856
```

```r
data("mtcars")  # Load mtcars dataset

# One-Sample T-Test on 'mpg'
oneSampleTTest(x = mtcars$mpg, mu = 20)
```

```
##
##    One sample t-test
##
## Data variable:   mtcars$mpg
##
## Descriptive statistics:
##                mpg
##    mean     20.091
##    std dev.  6.027
##
## Hypotheses:
##    null:        population mean equals 20
##    alternative: population mean not equal to 20
##
```

```
## Test results:
##    t-statistic:  0.085
##    degrees of freedom:  31
##    p-value:  0.933
##
## Other information:
##    two-sided 95% confidence interval:  [17.918, 22.264]
##    estimated effect size (Cohen's d):  0.015
```

```
# Bootstrap T-Test
boot.t.test(x = mtcars$mpg, mu = 20)
```

```
##
##  Bootstrap One Sample t-test
##
## data:  mtcars$mpg
## number of bootstrap samples:  9999
## bootstrap p-value = 0.9003
## bootstrap mean of x (SE) = 20.06177 (1.03862)
## 95 percent bootstrap percentile confidence interval:
##  18.10594 22.19063
##
## Results without bootstrap:
## t = 0.08506, df = 31, p-value = 0.9328
## alternative hypothesis: true mean is not equal to 20
## 95 percent confidence interval:
##  17.91768 22.26357
## sample estimates:
## mean of x
##  20.09062
```

```
data("iris")  # Load iris dataset

# Independent Samples T-Test between Sepal.Length for Species setosa and versicolor
iris_subset <- subset(iris, Species %in% c("setosa", "versicolor"))
independentSamplesTTest(formula = Sepal.Length ~ Species, data = iris_subset, var.equal = TRUE)
```

```
## Warning in independentSamplesTTest(formula = Sepal.Length ~ Species, data =
## iris_subset, : grouping variable has unused factor levels
```

```
##
##    Student's independent samples t-test
##
## Outcome variable:   Sepal.Length
## Grouping variable:  Species
##
## Descriptive statistics:
##             setosa versicolor
##    mean      5.006      5.936
##    std dev.  0.352      0.516
##
## Hypotheses:
```

```
##    null:        population means equal for both groups
##    alternative: different population means in each group
##
## Test results:
##    t-statistic:  -10.521
##    degrees of freedom:  98
##    p-value:  <.001
##
## Other information:
##    two-sided 95% confidence interval:  [-1.105, -0.755]
##    estimated effect size (Cohen's d):  2.104
```

```r
# Bootstrap T-Test for independent samples
boot.t.test(formula = Sepal.Length ~ Species, data = iris_subset)
```

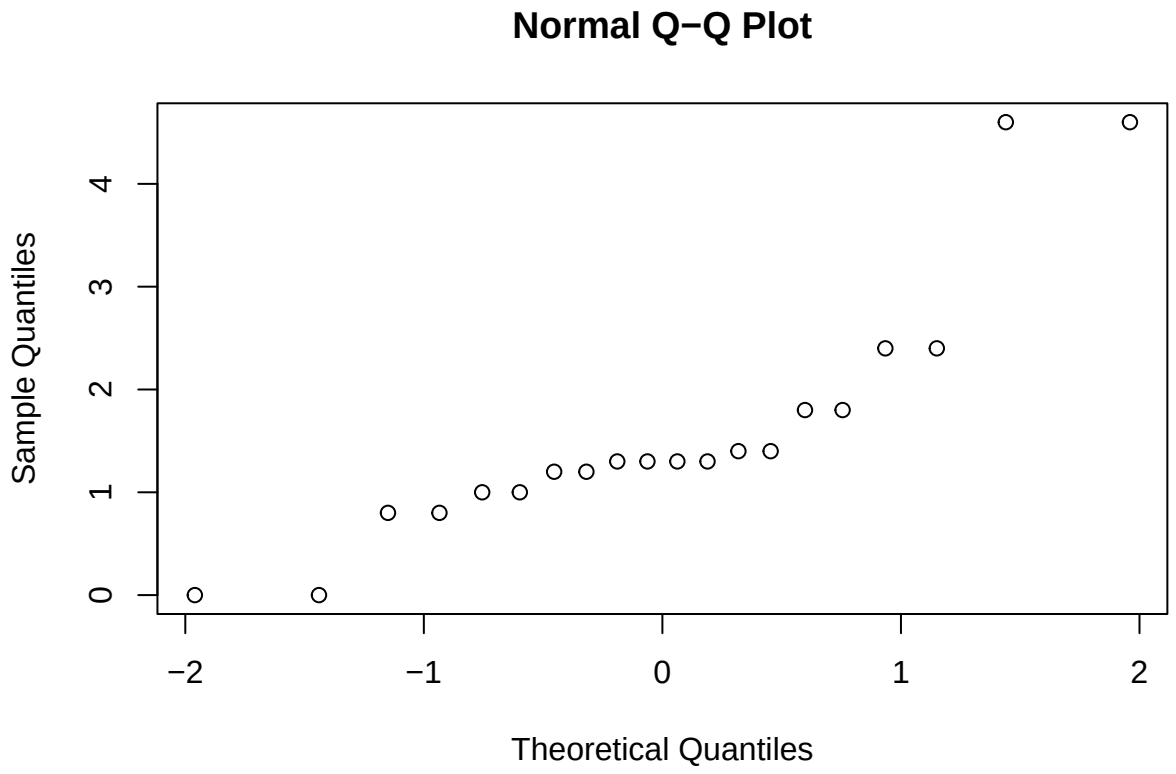
```
##
##  Bootstrap Welch Two Sample t-test
##
## data:  Sepal.Length by Species
## number of bootstrap samples:  9999
## bootstrap p-value < 1e-04
## bootstrap difference of means (SE) = -0.9300196 (0.08738447)
## 95 percent bootstrap percentile confidence interval:
##  -1.104 -0.758
##
## Results without bootstrap:
## t = -10.521, df = 86.538, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.1057074 -0.7542926
## sample estimates:
##     mean in group setosa mean in group versicolor
##                    5.006                    5.936
```

```r
# Calculate improvement in hours of extra sleep
sleep$improvement <- sleep$extra[sleep$group == 2] - sleep$extra[sleep$group == 1]
oneSampleTTest(sleep$improvement, mu = 0)
```

```
##
##    One sample t-test
##
## Data variable:   sleep$improvement
##
## Descriptive statistics:
##             improvement
##    mean           1.580
##    std dev.       1.197
##
## Hypotheses:
##    null:        population mean equals 0
##    alternative: population mean not equal to 0
##
## Test results:
```

```
##    t-statistic:  5.902
##    degrees of freedom:  19
##    p-value:  <.001
##
## Other information:
##    two-sided 95% confidence interval:  [1.02, 2.14]
##    estimated effect size (Cohen's d):  1.32
```

```r
# Check normality
qqnorm(sleep$improvement)
```




```r
shapiro.test(sleep$improvement)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sleep$improvement
## W = 0.80298, p-value = 0.0009573
```